УДК 004.052.42

Р.С. ЯРМОЛЮК
Хмельницький національний університет

УНІФІКАЦІЯ АТРИБУТІВ КОРТЕЖУ БАЗИ ДАНИХ ЗАСОБАМИ РЕГУЛЯРНИХ ВИРАЗІВ НА ПРИКЛАДІ ЕЛЕКТРОННОГО КАТАЛОГУ БІБЛІОТЕКИ

Розглянута одна з актуальних проблем: задача стандартизації та уніфікації записів електронного каталогу бібліотеки за допомогою апарату регулярних виразів. Проведено огляд та аналіз основних задач, що виникають при розробці програмних систем верифікації та очищення даних електронного каталогу.

Considered one of the actual problems: the task of standardization and unification of records electronic catalog of the library using the system of regular expressions. A review and analysis of the main problems that arise when developing software systems verification and data cleaning electronic catalog.

Ключові слова: регулярні вирази, бази даних, електронний каталог, стандартизація, уніфікація, розщеплення атрибутів.

Постановка проблеми

Однією з найбільш актуальних проблем у процесі організації роботи електронного каталогу бібліотеки являється реалізація функції забезпечення пошуку та корекції помилкових значень атрибутів бази даних електронного каталогу. Однак, рішення даної проблеми неможливе без ефективного механізму стандартизації та уніфікації структури запису у електронному каталозі. Оскільки під час роботи з базою даних електронного каталогу можуть виникати виключні ситуації, що призводять до спотворення, як самих значень атрибутів, так і їхньої структури запису.

Робота з аналізу та пошуку помилок в записах електронного каталогу передбачає роботу з великими масивами текстових даних. Такі поля бібліографічного запису, як авторський знак, УДК та ББК індекси, ISBN – номер, мають певну визначену шаблонну структуру запису, перевірка якої не завжди інтегрована у систему перевірки коректності системи керування базою даних (СКБД) електронного каталогу. З іншого боку, поширеною проблемою у роботі з базою даних електронного каталогу є NULL-значення (тобто відсутність даних) у певних полях бібліографічного запису. Для розв`язання даної проблеми відомий механізм запозичень інформації із зовнішніх джерел. До таких джерел можна віднести:

- інформаційні ресурси мережі Інтернет;
- списки використаних джерел та переліки посилань у базах даних наукових статей;
- бібліографічні описи із зовнішніх баз даних видавництв та бібліотек.

Якщо для імпорту даних із зовнішніх електронних каталогів існують інструментарії обробки у структурі автоматизованих бібліотечних інформаційних систем (АБІС), то для мережі Інтернет та списків літератури необхідно застосовувати підходи синтаксичного аналізу. Зокрема, розробка модулів синтаксичного аналізу HTML сторінок є досить трудомісткою роботою.

До основних задач, що виникають при розробці систем верифікації даних електронного каталогу бібліотеки можна віднести:

- Перевірка на коректність стандарту структурних (мають визначену структуру запису) атрибутів кортежу бази даних електронного каталогу. Індексні атрибути (авторський знак, УДК, ББК, ISBN, рік видання) мають наперед визначену структуру запису.
- Приведення даних до одного уніфікованого запису. Такі атрибути бібліографічного запису, як автор, назва, видавництво тощо мають довільну структуру запису, але для ефективного пошуку та обробки даних необхідно привести всі дані одного атрибуту до певного шаблону запису. Зазвичай, засобів самої СКБД недостатньо.
- Розбір бібліографічного опису на складові, або у загальному, виокремлення значень із атрибутів вільного формату (розщеплення атрибутів). Сам електронний каталог містить у собі велику кількість необробленої та неструктурованої інформації. Зокрема, списки використаних джерел та переліки посилань, що містяться у наукових статтях, монографіях, навчальних посібниках. Структура таких записів визначена відповідними нормативними документами. Тому пошук таких структур у тексті за певним шаблоном та виокремлення необхідних атрибутів для запису в базу даних є актуальною проблемою.
- Запозичення бібліографічної інформації з мережі Інтернет. Задачу доповнення відсутніх даних можливо вирішити за допомогою інформаційного поля мережі Інтернет. На основі відомої інформації формується пошуковий запит для певної пошукової машини (Google, Яндекс, Mail.ru тощо). Отриманий результат являє собою документ у розмітці HTML.

Аналіз останніх досліджень і публікацій

Проблемами пошуку та корекції помилок в записах електронного каталогу займалися такі вчені, як Вершинин М.И., Карауш А.С., Nielsen R., Ballard T., Randall B. та інші [1– 5]. У своїх працях дані автори приділяли увагу розробці методів та засобів пошуку та корекції орфографічних помилок у бібліографічних записах. Але питання стандартизації та уніфікації полів запису та розробки механізмів доповнення та уточнення відсутніх даних залишаються відкритими. З іншого боку, поставлену задачу можна віднести до однієї з проблем очистки даних при процесах побудови сховищ даних [6].

Формулювання цілей статті та актуальність досліджень

Основною метою даного дослідження є розробка ефективних методів та засобів розв'язання проблеми стандартизації та уніфікації структури запису значень певних атрибутів кортежу у базі даних електронного каталогу бібліотеки.

Аналіз можливостей сучасних АБІС показав відсутність відповідного інструментарію для забезпечення контролю над структурою запису атрибутів та для розв'язання задачі розщеплення атрибутів, що викликає багато труднощів для роботи коректора електронного каталогу. Зокрема, при роботі з електронним каталогом виникає потреба ручної перевірки записів. Тому розробка ефективних методів та засобів для забезпечення стандартизації та уніфікації структури запису в електронному каталозі бібліотеки є досить актуальною проблемою.

Виклад основних матеріалів дослідження

Формальний апарат регулярних виразів. Нехай $\Sigma = \{s_1, s_2, ..., s_3\}$ – алфавіт символів. Словом в алфавіті Σ називається послідовність записаних символів підряд, а довжиною слова – кількість його символів. Порожнє слово, що не містить символів, будемо позначати як e. Алфавіт Σ можливо розглядати як підмножину всіх слів довжини 1. Задамо операцію конкатенації алфавіту Σ самим з собою, результат якої дає множину всіх слів довжини 2 і позначається як $\Sigma\Sigma$ або Σ^2. Множину усіх слів довжини k будемо позначати – Σ^k і розглянемо її як k – кратну конкатенацію алфавіту Σ. Множину всіх непустих слів довільної довжини, що отримаємо об'єднанням всіх множин Σ^k, позначимо як Σ^+, а об'єднання цієї множини з пустим словом називається ітерацією мови і позначається як $\Sigma^* = \Sigma^+ \cup e$. Ітерація мови описує всі слова, які можливо побудувати в даному алфавіті. Довільна підмножина слів $L(\Sigma) \subset \Sigma^*$ називається формальною мовою в даному алфавіті Σ [7].

Визначимо тепер клас формальних мов, що задаються регулярними множинами. Регулярна множина задається рекурсією за правилами:

- порожня множина, а також множина, яка містить порожнє слово та одноелементні множини, які містять символи алфавіту, є регулярними базисними множинами;
- якщо множини P і Q являються регулярними, то множини, що побудовані застосуванням операцій об'єднання, конкатенації та ітерації, також є регулярними.

Регулярні вирази надають зручний спосіб запису регулярних множин у вигляді текстового рядка. Аналогічно множинам, регулярні вирази задаються рекурсією за правилами:

- регулярні базисні вирази задаються символами і визначають відповідні регулярні базисні множини;
- якщо p і q регулярні вирази, то результати операцій об'єднання, конкатенації та ітерації є також регулярними виразами, що задають відповідні регулярні множини.

Отже, регулярний вираз – це формальна мова пошуку і проведення маніпуляцій з текстовими рядками, заснована на використанні службових символів та символів-джокерів. Важливою практичною особливістю регулярних виразів є можливість побудови для довільного регулярного виразу скінченного автомату, що розпізнає приналежність заданого слова формальній мові, яка породжується регулярними виразами [7].

Синтаксис регулярних виразів залежить від платформи або мови програмування, на якій реалізується програмне забезпечення, та застосовує різний набір засобів, зокрема [8]:

- символи і escape-послідовності;
- символи операції і символи, що позначають спеціальні класи множин;
- імена груп і обернені посилання;
- символи тверджень та інші засоби.

На даний час переважна більшість програмних платформ та мов програмування реалізують у своїх можливостях підтримку регулярних виразів [8]. Серед них Perl, Java, PHP, JavaScript, мови платформи.NET Framework, Python, Ruby, та інші. Наприклад, для об'єктно-орієнтованої мови програмування C# дані можливості реалізовані у просторі імен System.Text.RegularExpressions.

Уніфікація атрибутів з наперед визначеною структурою. Розглянемо задачу структурної верифікації значень стандартизованих атрибутів кортежів бази даних електронного каталогу. Нехай $\Sigma_{atr(1)}, \Sigma_{atr(2)}, ..., \Sigma_{atr(n)}$ – повні алфавіти відповідних стандартизованих атрибутів, тобто атрибутів, структура запису яких регулюється наперед відомими нормами. Побудова алгоритмів відбувається на основі правил та стандартів допустимих символів у значеннях атрибуту. Також на основі норм та допустимих форматів запису значення атрибуту, побудуємо формальні мови $L(\Sigma_{atr(1)}), L(\Sigma_{atr(2)}), ..., L(\Sigma_{atr(n)})$ та відповідні множини регулярних виразів, що задають їх $R_{atr(1)}, R_{atr(2)}, ..., R_{atr(n)}$, де $R_{atr(i)} = \bigcup_{j=1}^{m} R_{atr(i)}^{(j)}$ – множина регулярних виразів усіх допустимих правил побудови формату запису і-го атрибуту. Оскільки, ми

розглядаємо значення кожного атрибуту, як регулярний вираз, що задає відповідну формальну мову, то існує такий скінченний автомат акцептор (FSM), на вході якого задається запис атрибуту ($word$), а на виході продукує $true$, якщо слово належить формальній мові, інакше $false$:

$$FSM[R_{atr(i)}](word) = \begin{cases} true, \ word \in L(\Sigma_{art(i)}) \\ false, \ word \notin L(\Sigma_{art(i)}) \end{cases}. \quad (1)$$

Розглянемо випадок $m = 1$, тобто існує єдиний регулярний вираз, що задає регулярну множину значень певного і-го атрибуту $R_{atr(i)} = R^{(1)}_{atr(i)}$. Запропонуємо загальну схему перевірки на коректність записів і-го атрибуту у базах даних електронного каталогу:

1. На основі стандартів та правил запису *i*-го атрибуту записуємо регулярний вираз $R_{atr(i)}$.

2. Кожне значення атрибуту у циклі перевіряється на приналежність до множини коректних значень за допомогою скінченного автомата акцептора $FSM[R_{atr(i)}]$.

3. Значення атрибуту, що не пройшли перевірку на коректність, переадресовуються до модуля виправлення помилок, або перевіряються коректором.

У випадках, коли правил та стандартів запису декілька $m > 1$, (наприклад, коли у різні роки діяли різні вимоги щодо оформлення, або допускаються декілька можливих варіантів запису) перевірка на коректність відбувається для кожного з регулярних виразів, що відображають певну множину правил або стандартів.

Введемо поняття потужності регулярного виразу відносно домену *i*-го атрибута $U_{atr(i)} = dom(atr(i))$. Під потужністю регулярного виразу будемо розуміти кількість значень атрибуту, які коректні за регулярним виразом:

$$\left|R_{atr(i)}\right| = \left|U_{atr(i)} \ \mathbf{I} \ L(\Sigma_{atr(i)})\right|. \quad (2)$$

Уніфікація атрибутів невизначеної структури. Розглянемо випадок, коли правила та стандарти для запису значень атрибутів або відсутні, або дозволяють записи довільного формату (автор, назва, видавництво тощо). У таких випадках неможливо однозначно перевірити на коректність запис атрибуту, адже множина усіх регулярний виразів для даного атрибуту є невизначеною. Щоб позбутися даної невизначеності, запропонуємо ітераційний алгоритм для побудови множини регулярних виразів для певного атрибуту.

0-ітерація. Відокремлюємо деяку регулярну підмножину із загальної множини всіх значень *i*-го атрибуту $U_{atr(i)}$ та на її основі будуємо регулярний вираз $R^{(1)}_{atr(i)}$. Тоді $R_{atr(i)} = R^{(1)}_{atr(i)}$.

1-ітерація. Всі елементи множини $U_{atr(i)}$ перевіряються на коректність за регулярним виразом $R_{atr(i)}$. Якщо $\left|U_{atr(i)}\right| - \left|R_{atr(i)}\right| = 0$ процес завершується, інакше відокремлюємо нову регулярну підмножину із множини $U_{atr(i)} \setminus R_{atr(i)}$, та на її основі будуємо регулярний вираз $R^{(2)}_{atr(i)}$. Тоді $R_{atr(i)} = R^{(1)}_{atr(i)} \ \mathbf{U} \ R^{(2)}_{atr(i)}$ і так далі.

Запишемо загальний вигляд множини регулярного виразу для *j*-го кроку:

$$R_{atr(i)} = \begin{cases} R_{atr(i)} \ \mathbf{U} \ R^{(j)}_{atr(i)}, & \left|U_{atr(i)}\right| - \left|R_{atr(i)}\right| \neq 0 \\ R_{atr(i)}, & \left|U_{atr(i)}\right| - \left|R_{atr(i)}\right| = 0 \end{cases}, \quad (3)$$

де $R^{(j)}_{atr(i)}$ – регулярний вираз побудований для регулярної підмножини з множини $U_{atr(i)} \setminus \mathop{\mathbf{U}}\limits_{k=1}^{j-1} R^{(k)}_{atr(i)}$.

Задача приведення атрибутів до одного уніфікованого формату. У випадках, коли відсутні стандарти та норми запису значень атрибутів і прийняття рішення про приведення даних до уніфікованої форми покладається на користувача, виникає проблема вибору базового формату запису атрибуту, тобто регулярного виразу, за допомого якою проводити уніфікацію атрибуту. Оптимальний базовий формат запису *i*-го атрибуту ($R^{base}_{atr(i)}$) обирається з переліку регулярних виразів, що отриманий при застосуванні ітераційного алгоритму (3). Задамо критерій оптимальності, як максимум із потужностей заданих регулярних виразів:

$$\max_{j=1..m} \left|R^{(j)}_{atr(i)}\right| \Leftrightarrow \left[R^{base}_{atr(i)} = R^{(j)}_{atr(i)}\right] \quad (4)$$

Даний критерій оптимальності мінімізує кількісні витрати на приведення всіх інших форматів до

базисного. Маючи базовий регулярний вираз, визначимо процедури перетворення (CF), як набір правил перетворення формату і-го атрибуту, заснованих на іменах груп та обернених посиланнях:

$$CF_{atr(i)}^{(j)to(base)}(word) = [R_{atr(i)}^{j} \Rightarrow R_{art(i)}^{base}](word) \quad (5)$$

Застосувавши послідовно до кожного елемента домену *i*-го атрибуту відповідну процедуру перетворення (5), отримаємо значення атрибуту, що повністю приведені до уніфікованої форми. На даний момент залишається відкритою проблема щодо можливості побудови процедур перетворення. Для розв`язання даної проблеми доведемо теорему.

Теорема. Для будь-яких двох регулярних виразів (з множини допустимих) для *i*-го атрибуту можливо побудувати хоча б одну процедуру перетворення, або

$$\forall\ (R_{art(i)}^{j} \subset U_{atr(i)}) \wedge (R_{art(i)}^{k} \subset U_{atr(i)}),\ \exists\ CF_{atr(i)}^{(j)to(k)}. \quad (6)$$

Доведення. Доводити будемо від протилежного. Нехай неможливо побудувати процедуру перетворення $CF_{atr(i)}^{(j)to(k)}$ для і-го атрибуту. Тоді відповідні алфавіти, що реалізовані в регулярних виразах, мають хоча б 1 відмінний символ, тобто $\exists\ \Sigma_{atr(i)}^{'},\ \Sigma_{atr(i)}^{'} \cup \Sigma_{atr(i)} \neq \Sigma_{atr(i)}$. З іншого боку, оскільки $\Sigma_{atr(i)}$ є повним алфавітом і-го атрибута, то $\Sigma_{atr(i)}^{'} \subset \Sigma_{atr(i)} \Leftrightarrow \Sigma_{atr(i)}^{'} \cup \Sigma_{atr(i)} \equiv \Sigma_{atr(i)}$. Дане протиріччя і доводить теорему.

Розщеплення атрибутів. Атрибути вільного формату найчастіше містять у собі множини окремих значень, що підглядають розщепленню для підвищення точності відображення і підтримки наступних етапів обробки даних. Типовою проблемою є проблема розщеплення атрибутів бібліографічного опису для подальшого використання у процесах запозичення значень. Дана проблема вирішується апаратом регулярних виразів. На першому етапі розробляються регулярні вирази, що задають стандартизовані правила бібліографічного запису (ДСТУ ГОСТ 7.1: 2006 "Система стандартів з інформації, бібліотечної та видавничої справи. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання"). Далі, за допомогою алгоритмів розпізнавання за регулярними виразами, з тексту виокремлюються рядки, що є бібліографічними описами та застосовуються методи розщеплення даних на атрибути.

Висновки

Запропоновані у даній роботі підходи щодо уніфікації та стандартизації атрибутів запису електронного каталогу можуть бути використані при побудові програмних комплексів систем верифікації баз даних та телекомунікаційних технологій. Теорія формальних мов та апарат регулярних виразів легко реалізуються за допомогою сучасних мов програмування, зокрема, для об'єктно-орієнтованої мови програмування C# дані можливості реалізовані у просторі імен System.Text.RegularExpressions. У подальшому даний підхід можливо застосувати для задач синтаксичного аналізу HTML-сторінок у задачі запозичень даних із мережі Інтернет.

Література

1. Вершинин М.И. Электронный каталог проблемы и решения / Вершинин М.И. – СПб. : ПРОФЕССИЯ, 2007. – 233 с.

2. Карауш А.С. Утилиты для проверки и коррекции электронных каталогов / Карауш А.С., Копытков Д.Ю., Макаревич А.С. // Библиотечное дело. – 2005. – № 6 (30). – С. 21–24.

3. Nielsen R. Lost articles: Filing problems with initial articles in data bases / R. Nielsen, J. M. Pyle // Libr. Resources a. techn. Services. – 1995. – Vol. 39, №3. – P. 291 – 293.

4. Ballard T. Spelling and typographical errors in library databases: one libr. System for noting out spelling error / T. Ballard // Computer in libr. – 1992 – Vol. 12, № 6. – P. 14–19.

5. Randall B. Spelling Errors in the Database: Shadow or Substance? / B. Randall // Libr. Resources a. techn. Services. – 1999. – Vol. 43, №3. – P. 161–170.

6. Rahm E. Data Cleaning: Problems and Current Approaches / E. Rahm, H. D. Hong // IEEE Data Engineering Bulletin. – 2000. – Vol. 23, №4. – P. 3–13.

7. Пентус А.Е. Теория формальных языков : [учебное пособие] / А.Е. Пентус, М.Р. Пентус. – М. : МГУ, 2004. – 80 с.

8. Friedl J. Mastering Regular Expressions Third Edition / J. Friedl. – California: O'Reilly Media, 2006. – 534p.

Надійшла 21.12.2011 р.
Рецензент: д.т.н. Сорокатий Р.В.